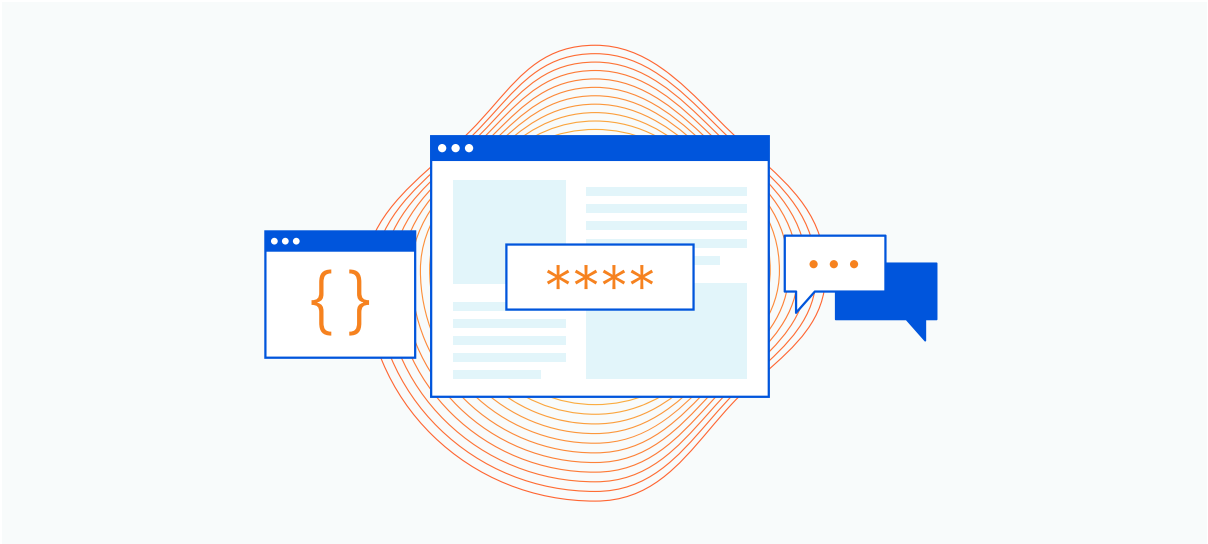
백서

CLOUDFLARE

API 보안 가이드

+82 70 4515 6893 | enterprise@cloudflare.com | www.cloudflare.com

API는 (앱) 세계의 원동력입니다



우리는 모두 API, 즉 응용 프로그램 프로그래밍 인터페이스가 세계의 원동력이라는 점을 알고 있습니다. 더 정확히 말해, API는 별개의 현대 응용 프로그램이 서로 통신할 수 있게 해줍니다. 모바일 또는 웹 응용 프로그램은 데이터를 저장하고 처리하는 백엔드에 액세스할 수 있습니다. API로 서로 다른 기업의 응용 프로그램이 공개적으로 통신할 수 있고, 더욱 흔하게는 비공개적으로 통신하여 비즈니스 목표를 충족하도록 내부 응용 프로그램을 통합합니다.

그 결과는 어떨까요? 더욱 폭넓은 기능과 다양한 데이터로 응용 프로그램, 웹 사이트, 모바일 앱이 더욱 강력하고 완전해집니다.

예를 들어, 차량 공유 회사는 처음부터 자체 결제 서비스를 구축하는 대신 결제 회사의 API로 결제 서비스를 추가할 수 있습니다. 또 하나의 예는 항공편 통합 사이트입니다. 이런 사이트에서는 항공편 시간, 가격, 목적지 등 비행기 티켓에 대해 알아야 할 모든 내용을 표시하기 위해 API 호출로 항공사 데이터베이스에 연결하여 적절한 데이터를 가져오고 통합 검색 결과 페이지에 표시합니다.

“API 우선”이라고 내세우는 회사들이 점점 늘어나면서 API의 중요성이 커지고 있습니다. 회사의 실제 제품이 이를 소비하는 비즈니스 모델에 중점을 둔 API인 경우도 있습니다. 예를 들어 날씨 데이터를 제공하는 회사가 API를 제품으로 만들 경우, 날씨 정보가 필요한 다른 회사가 이 API에 액세스하는 데 월 요금을 지불하게 됩니다. 응용 프로그램이 다른 응용 프로그램과 상호작용해야 한다는 기대를 고려하여, 실제 제품 기능을 제공하는 코드와 함께 API를 설계하거나 심지어는 해당 코드 이상으로 설계하여 API가 개발 최종 단계에 고정되지 않는 경우가 많습니다.

회사에서는 API가 적절한 데이터를 노출해 수익 및 비즈니스 모델의 기반이 될 방법을 고려하기 위하여, API에 접근할 방식을 시간과 노력을 들여 신중히 구축합니다.

하지만 완벽한 API를 구축하기는 힘듭니다. 다른 소프트웨어와 마찬가지로 취약점이 생겨, 이 문서에서 다룰 보안 문제로 이어지기 때문입니다.

일단은 API가 모든 곳에 있으며 향후 몇 년 동안 더욱 성장할 것이고, 보호가 필요하다는 점까지만 설명하겠습니다. 이러한 이유로 조직의 API 보안을 고려한 API 공격 및 심층 방어 측면을 살펴보려고 합니다.

숫자로 살펴보는 API의 성장세

50%

Cloudflare를 통과하는 트래픽 중 API 트래픽의 비율

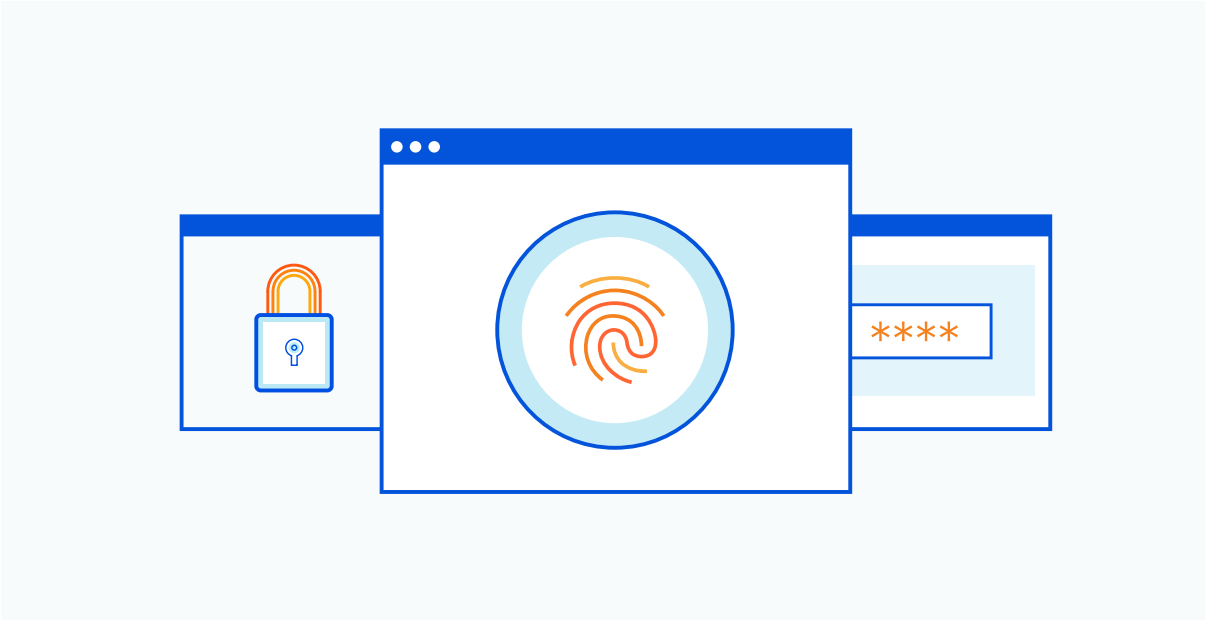
61%

해마다 증가하는 API 트래픽 비율

Programmable Web[1]에 따르면 게시되고 잘 알려진 API는 24,000여 개가 있습니다. 하지만 대부분의 API는 비공개이며, 내부 응용 프로그램을 서로 연결하는 것으로 나타납니다. 비공개 API의 수는 수백만 개로 추정됩니다.

Cloudflare에서는 API가 늘어나고 있다고 말하는 데 그치지 않고 API의 성장세를 직접 목격하고 있습니다. Cloudflare Radar의 2021년 상반기 데이터에 따르면 Cloudflare 네트워크 트래픽의 절반가량이 API와 관련되었습니다. 심지어 2020년에서 2021까지는 61퍼센트 증가했습니다.

Cloudflare는 API에서 중요한 데이터가 노출된다는 점을 고려하여, 보호해야 할 막대한 공격 표면을 API가 어떻게 새로 드러내는지 알 수 있습니다. 어떻게 알 수 있을까요? 최근 몇 년간 API를 겨냥한 공격이 눈에 띄게 많았습니다.



[1]https://www.programmableweb.com/apis/directory

공격 표면의 확장

Cloudflare는 API가 모든 곳에 있으며 현대 비즈니스의 성공에 기본적인 요소라는 점을 인식하고 있습니다. API는 응용 프로그램 로직을 노출하며 다른 응용 프로그램과 중요한 데이터를 공유할 수 있습니다.

하지만 당연하게도, 공격자가 이 점을 알고 있으며 이와 같이 확장되는 기업의 공격 표면을 악용하려는 의도가 충분하다는 점이 드러났습니다.

Gartner가 단언한 다음과 같은 내용이 옳을 수도 있습니다.[2] 2022년이면 API 남용은 "빈번하지 않았던 공격 벡터에서 가장 빈번한 공격 벡터로 옮겨가서 기업 웹 응용 프로그램의 데이터 유출 원인이 될 것이다."

API가 가장 자주 공격 벡터의 대상이 될지는 아직 두고 봐야 하겠지만, 공격자가 API를 계속 겨냥할 것은 분명합니다.

"계속"이라고 표현한 이유는 API 보안이 취약하거나 보안을 고려하지 않고 API를 개발했기 때문에 세간의 이목을 끌 정도로 유출이 발생한 사례가 전 세계적으로 이미 목격되었기 때문입니다.

T-Mobile은 2017년에 새 기기를 구매했거나 T-Mobile 계정을 신청한 1,500백만 고객의 정보가 노출되면서 API 공격의 희생양이 되었습니다. 데이터에는 이름, 주소, 생년월일, 사회보장번호, 운전면허증 번호, 여권 번호가 포함되었습니다. 부사장이 보고한 바에 따르면 공격은 API 호출의 전화번호 파라미터를 조정하여 수행되었습니다. 모든 사용자의 전화번호를 쿼리할 수 있었으며 T-Mobile API는 전화번호가 쿼리된 사람의 중요한 계정 데이터를 포함한 응답을 보내게 되었습니다.

또 다른 API 관련 유출이 USPS를 강타했습니다. 실시간 패키지 추적을 지원하는 API에 기본 인증이 부족한 것으로 확인되었습니다. 로그인한 사용자는 데이터 집합의 모든 레코드를 가져다주는 와일드카드 검색 파라미터를 사용해 다른 모든 사용자의 계정 정보를 쿼리할 수 있었습니다. 이로 인해 6천만 USPS 계정 소유자가 위험에 처하게 되었습니다.

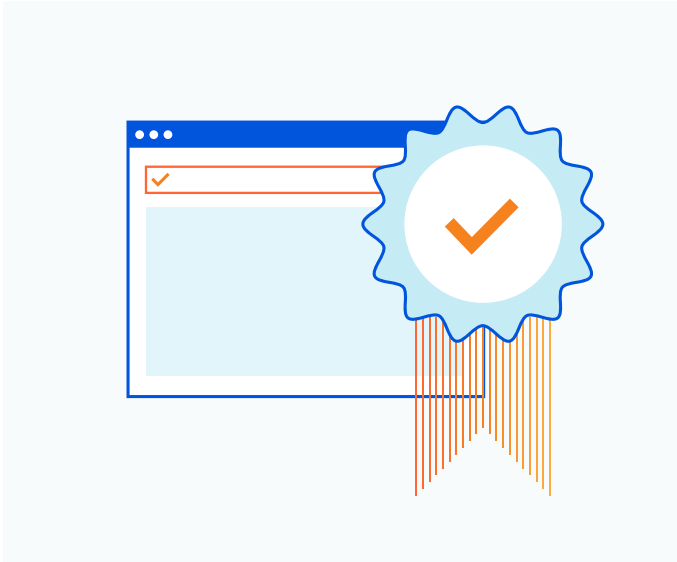
2019년에 인도 현지의 대형 검색 엔진인 JustDial에서는 API를 새 버전으로 교체하면서 이전 버전을 남겼다가 이후 노출되어 사실상 모든 고객 데이터가 유출되었습니다. 상황은 이후 악화됐는데, 사실상 인증 방법이 마련되어 있지 않아 누구든 API를 호출하여 프로덕션 서버에서 데이터를 스크래핑할 수 있었습니다. 다시 말해, 사용자 데이터에 액세스하는 데 고급 기법이 필요하지 않았습니다.

Facebook에서도 마찬가지로, Facebook이 GraphQL을 이끄는데도 불구하고 API로 인해 여러 번 유출이 발생했습니다. 예를 들어 2019년 말 Facebook에서는 데이터베이스가 스크래핑되어 사용자 2억 6천만 이상의 이름, 전화번호, 사용자 ID가 위험에 처했습니다.

API를 보호하는 데 어려움을 겪은 조직의 목록은 길게 이어집니다. 이유는 여러 가지입니다. 먼저, 설계가 안전하지 않아 발생하는 근본적인 API 취약점 때문에 공격 기회가 생깁니다. 게다가 지금까지 조직에는 API 우선 보안 도구가 없었습니다. WAF나 속도 제한과 같은 웹 보안 도구를 이용할 수는 있었지만, 이런 도구는 API가 아니라 응용 프로그램을 보호하도록 구축되어 있었습니다. 이로 인해 긍정 오류 등의 어려움이 발생할 수 있으며, 대규모로 자동화된 트래픽에 맞게 설계된 API 보안이 분명 필요합니다.

[2]출처: Gartner: "API Security: What You Need to Do to Protect Your APIs", 2021년 3월

리믹스! 새로운 OWASP API 상위 10가지



이 모든 상황에서 한 가지 희망은 오랫동안 응용 프로그램 보안을 개선하는 데 주력해 온 조직인 OWASP 재단이 참여했다는 점입니다. OWASP는 웹 응용 프로그램 보안 위험 상위 10가지로 이름이 알려졌으며, 중요도가 높은 API 보안 위험과 취약점 목록인 API 보안 상위 10가지를 현재 게시하고 있습니다.

사실, 응용 프로그램 보안에 대해 Cloudflare에서 우려하는 부분은 API를 구축하고 보호하는 데도 마찬가지로 우려됩니다.

API를 우선하는 모든 조직에서는 처음에 API를 설계할 때 보안을 먼저 고려해야 합니다. 방금 언급한 몇 가지 공격과 악용되는 OWASP 보안 위험을 살펴보겠습니다.

OWASP API 상위 10가지

1. 손상된 개체 수준 인가
2. 손상된 사용자 인증
3. 과도한 데이터 노출
4. 리소스 부족 및 속도 제한
5. 손상된 기능 수준 인가
6. 대량 할당
7. 잘못된 보안 구성
8. 인젝션 (Injection)
9. 부적절한 자산 관리
10. 불충분한 로깅 및 모니터링

주요 API 보안 문제점

1. 손상된 인증 및 권한 부여

앞서 언급한 공격에서 악용한 몇 가지 주요 OWASP API 위험에 대해, 인증과 권한 부여부터 자세히 살펴보겠습니다.

JustDial은 엔드포인트에서 인증이 손상되어 무너졌고, 모든 사람이 호출할 수 있게 되었습니다. 인증이 구현되면 올바른 TLS 인증서, API 키, 웹 토큰 등이 있는 API 호출만 요청할 수 있어 API 보안 위험이 상당히 줄어듭니다.

USPS와 T-Mobile 사례에서 확인했던 것처럼, 다수의 API 공격은 OWASP 목록의 상위 1위 요인에 따라 인증이 취약하거나, 손상되었거나, 마련되어 있지 않은 점을 악용합니다. 개체 수준 권한 부여가 손상되는 경우는 흔합니다. 액세스하도록 권한이 부여된 개체의 ID 번호를 액세스하도록 권한이 부여되지 않은 ID 등으로 대체하는 방식으로 API 엔드포인트가 악용됩니다. 요청 시 개체 ID만 바꿔도 데이터에 무단 액세스할 수 있는 경우가 흔합니다.

API 경로 및 쿼리 파라미터에는 액세스하고 있는 리소스 ID가 포함됩니다.

권한이 부여된 호출:

`GET api.greatsampleapis.com/v1/users/235` 사용자 ID가 235인 경우.

조작된 API 호출은 235를 236으로 바꾸어 무단으로 액세스할 수 있게 됩니다. 즉, 이 사례에서는 사용자 ID인 개체 식별자를 조정하여 사용자 236에 대한 데이터에 액세스하는 것입니다.

`GET api.greatsampleapis.com/v1/users/236`

권한 부여 관리가 마련되어 있지 않다면 이 방법은 성공할 수 있습니다. 개발자는 공격으로 개체 ID 값을 조정하거나 수정하여 다른 데이터에 액세스 권한을 얻지 못하도록 엔드포인트에 위협을 모델링해야 합니다. 예측할 수 없는 개체 ID 값을 사용하는 것도 순차적이지 않고 쉽게 추측할 수 없으므로 도움이 됩니다.

2. 대량 할당, 데이터 노출, 삽입 공격

또 다른 공격 계층으로는 응답에 지나치게 많은 데이터가 노출되거나 입력으로 내부 개체를 조정할 수 있게 됩니다.

API가 개체 속성을 광범위하게 노출하려 하고 응답으로 데이터가 지나치게 많이 반환될 때, 데이터를 필터링하도록 요청하는 클라이언트를 이용하게 되면서 데이터가 과도하게 노출됩니다.

공격자는 응답의 추가적인 세부 정보를 이용해 더 강력하게 공격하거나 피싱 이메일을 만들어낼 수도 있습니다. 예를 들어 응답에서 아래와 같은 데이터를 모두 반환하는 경우, 아주 그럴듯해 보이는 피싱 이메일을 만드는 데 해당 데이터가 사용될 수도 있습니다.

```
{
“Id”: 213,
“FirstName”:“Sanjay”,
“LastName”:“Smythe”,
“EmailAddress”:“ssmythe@hacketyhack.com”,
“Occupation”:“Assistant to the Deputy Associate Vice Sub-undersecretary",
“DOB”:“1986-05-21”,
“Bank”:“Easygo Financial”,
“AccountNumber”: 1362886306,
“PetName”: “Aloysius”,
}
```

API가 내부 개체와 변수를 노출할 경우 대량 할당 공격을 수행해 API 호출로 내부 값을 변경하거나 악용할 수 있습니다.

OWASP는 이렇게 표현합니다.

> “때때로 프레임워크를 더욱 쉽게 사용하도록, 개발자가 소프트웨어 프레임워크로 HTTP 요청 파라미터를 프로그램 코드 변수나 개체에 자동 바인딩할 때가 있습니다... 공격자가 이 방법을 이용해 개발자의 의도가 아닌 새 파라미터를 생성하여 의도되지 않은 새 변수나 개체를 프로그램 코드에 대신 생성하거나 덮어쓸 수 있게 되는 경우도 있습니다.”

개발자는 어떻게 해야 할까요? 개발 시 대량 할당을 수행하려 할 때 잠재적인 위험을 이해해야 하고 악용될 수 있는 내부 개체나 변수가 노출되지 않도록 제어해야 합니다. 클라이언트가 업데이트할 수 있는 허용 목록 속성도 고려해야 합니다.

웹 응용 프로그램은 오랫동안 삽입 공격에 취약했고, API도 마찬가지입니다. 삽입 공격은 아주 잘 알려져 있으므로 이 공격 방식을 자세히 설명하지는 않겠습니다. 입력을 전달하기 전에 검증하고 깨끗하게 처리해야 한다는 설명이면 충분할 것입니다. API 응답에 대한 데이터 유출 방지를 사용하고 반환될 수 있는 레코드 수를 제한하여 대량 유출 사고를 방지하도록 노력해야 합니다.

3. 리소스 남용 및 섀도우/로그 API

또 다른 공격 방식으로 API를 남용할 수 있습니다. 컴퓨팅 리소스를 과도하게 소모하여 DoS 방식의 공격으로 지배하고 무너지게 만드는 것입니다. 클라이언트/리소스당 요청 수, 단일 응답으로 반환되는 레코드나 요청 페이로드 크기와 같은 요소를 제한하지 않으면 이러한 공격을 받게 될 수 있습니다.

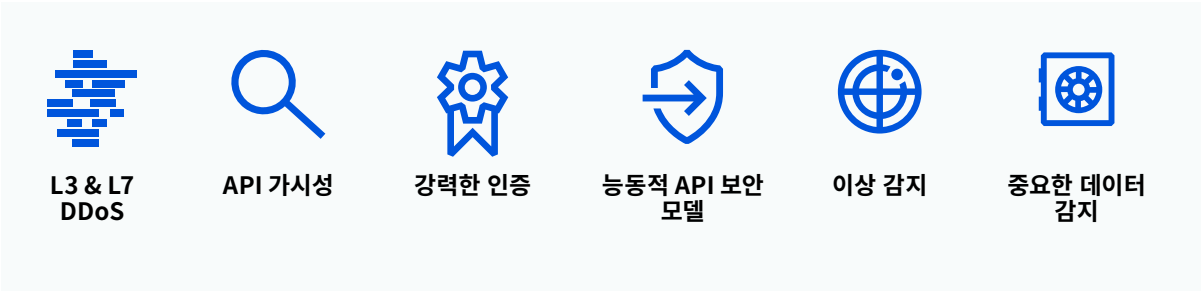
JustDial 공격에서 확인한 것처럼, 프로덕션 API는 잊혀져서 보호되지 않아 악용될 가능성이 높으므로 섀도우나 로그 API가 될 수 있습니다. 나머지 보안과 마찬가지로 IT 자산이나 공격 표면에 가시성을 확보해 적절한 보안 제어를 적용해야 합니다. API 엔드포인트 자산 전체에 가시성을 확보하는 것도 마찬가지입니다.

API를 개발하는 경우, API 버전을 추적하여 프로덕션 과정에 있는 API와 더 이상 사용되지 않는 API를 알 수 있도록 팀에서 세밀한 프로세스를 마련해야 합니다.

API 보호 고려 사항

Cloudflare에서는 API의 의미, API가 중요한 이유, API를 대상으로 하는 일반적 공격에 대해 확인했습니다. 이제 Cloudflare가 API 보안을 구축해 가장 흔한 공격으로부터 API를 보호하는 방식을 살펴보겠습니다. API 보안이 효과적이려면 가시성에서부터 능동적 보안 모델, 남용 중지, 데이터 보호에 이르기까지 모든 사항을 고려해야 합니다.

Cloudflare API Shield



가시성의 토대

가시성

보안의 다른 측면과 마찬가지로, 무언가를 보호하려면 볼 수 있어야 합니다. API도 마찬가지이며, 특히 회사에 수백 개, 심지어 수천 개의 API 엔드포인트가 있을 때는 더욱 볼 수 있어야 합니다.

API 탐색 및 가시성은 API를 관리하는 데 있어 핵심 측면이므로 조직에서는 항상 API 엔드포인트 자산에 대한 명확한 정보를 갖춰 섀도우 API나 로그 API로 문제가 발생하지 않도록 해야 합니다.

JustDial에서 확인한 것처럼, 조직에서 API를 추적하지 못할 경우 데이터 유출로 이어질 수 있습니다.

API 심층방어

API L7 보호

Cloudflare는 계층 7 DDoS 공격에 맞서 응용 프로그램을 보호하기 위해 오랫동안 웹 애플리케이션 방화벽을 배포했습니다. 속도 제한 및 DDoS 방어 등 여러 신뢰할 수 있는 제어 방법으로 API 보호를 시작해 서비스 거부 공격 및 무차별 암호 대입 로그인 시도와 특정 IP 주소로 수행되는 일반적인 남용을 방지해야 합니다. 이렇게 하면 API 사용량을 제한해 OWASP API의 4번째 요소인 리소스 부족 및 속도 제한을 방지할 수 있게 됩니다.

이와 더불어 API를 대상으로 하는 일반적인 공격을 구분하고 차단하는 데 WAF 규칙을 사용할 수도 있습니다.

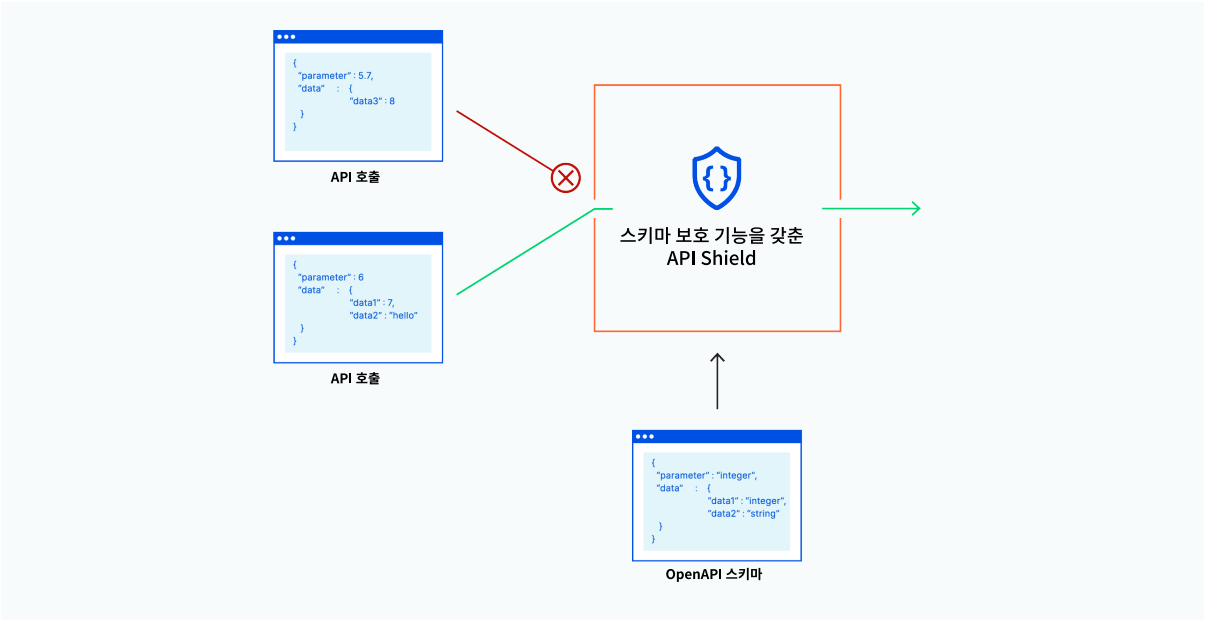
인증 및 권한 부여

mTLS 인증

API 공격을 설명하면서 인증이 부족하면 엄청난 피해를 입을 수 있다는 것을 확인했습니다. 처음부터 기본으로 인증을 갖추어야 하며 상호 TLS로 보완하여 모바일이나 IoT 등의 사용 사례에 인증서 기반 ID를 시행할 수 있어야 합니다. 이러한 접근 방식은 더욱 능동적인 허용 목록 모델로, 유효한 인증서가 있는 합법적 클라이언트가 보낸 요청만 연결하도록 허용해줍니다.

노출된 자격 증명 확인

API는 도난된 자격 증명을 사용하여 로그인 시도를 이어가는 자격 증명 스터핑 공격의 영향을 받습니다. 조직에서 통제할 수 없는 타사에서 벌어진 유출로 인해 계정의 자격 증명이 손상될 수도 있습니다. 인증 확인의 일부로서 API 보안은 유출된 자격 증명 데이터베이스와 대조해 로그인 시 인증 자격 증명을 스캔할 수 있어야 합니다. 자격 증명이 손상된 것 같으면, API 보안은 비밀번호 초기화나 다단계 인증 등 추가 보안 조치를 동원해야 하며 물론 해당 시도를 차단해야 합니다.



스키마 유효성 검사를 이용하면 각 요청을 API 스키마 로깅이나 이를 준수하지 않아 차단한 요청에 대해 평가할 수 있습니다.

능동적 API 보호

API 스키마 유효성 검사

개발자들은 API 스키마를 만들기 위해 최선의 노력을 다합니다. API 스키마란 API와 상호작용하도록 예상한 방식을 다룬 문서 또는 기본 규칙입니다. 각 엔드포인트(GET /users, POST /users)에서의 요청 메서드와 작업, 혹은 각 작업에 대한 입력 파라미터와 출력 파라미터 등을 설정할 수 있습니다. 흔히 Swagger 표준으로 알려지기도 한 OpenAPI v3는 API 정의용으로 가장 널리 알려진 스키마입니다.

신뢰할 수 있는 API 보안은 스키마 시행에 능동적인 제로 트러스트 모델을 이용해야 합니다.

스키마가 있다면 스키마에 대해 자동으로 요청의 유효성을 검사해야 합니다. 스키마를 준수하는 것으로 검증된 API 작업을 제외한 모든 API 작업은 차단됩니다.